UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart System Newsletter

24 October 1978 Vol. 4, No. 20
Send all comments, criticisms and contributions to the editor: T. W. Lanzatella
University Computer Center, 2520 Broadway Drive, Lauderdale, MN 55113
The University of Minnesota is an equal opportunity educator and employer.

TABLE OF CONTENTS

NOTICE OF CHANGES TO THE SYSTEM

NOS Changes

The next new system will be installed on Thursday, 26 October.

Kevin Matthews repaired shared queue file problems in QFM and SET and installed new versions of OEF and UQM.

Don Mears repaired a TID processing error in DSP and corrected a long standing error in SEND (a correct version has been on KLUDGES for about six weeks).

Tim Salo added code to SUPIO to enforce a page limit.

PROPOSED CHANGES TO THE SYTEM

Indexed Scanning of the Callprg Index - by M. Riviere

I am considering a modification to Callprg to make it search an index according to a directory. This modification will imply a change in the structure of the Callprg index, however, the capability of CALLPRG to search an index of the current format will not be removed. That is, private Callprg indices will not have to be modified. If the System Callprg index happens, at any time, not to have a directory, CALLPRG will also be able to scan it. (This could be in the case when the index had to be modified due to an emergency situation and the process of providing it with a new directory happens to be omitted or forgotten.)

The structure of an index directory should be the following:

Directory Records: These records start with a word with zero characters in the left side. (The current indices start with blank or entry lines.) The zero characters let CALLPRG know that it is reading an index directory. The following words are sets of the words tables, each table associated with an index entry. These tables contain entry names, the sector number where the record to which they belong is located within the file and the word's location within the record for the first word of the entry. This information allows CALLPRG to position the file and read directly the requested entry.

Index Entry Records: These records are composed by entry lines similar to the ones in the actual Callprg index. Comment lines are not included and the index entries are listed alphabetically. Each of these records is under 800 CM words.

I ran some tests to estimate the resources utilized by CALLPRG using two kinds of scan. I ran the tests in KCL loops, executed 500 times and I obtained the job's cost output for comparisons. That is, I retrieved a product or requested a non-existing product 500 times in order to compensate for the time and resources used by the rest of the job statements.

All tests were run on the Cyber 172 during System's time with almost no other machine activity.

Following is a list of the times and resources utilized. Considering that CALLPRG is used approximately 1000 times a day, without counting the "Not Found" cases, and that the change will save, according to the results which I obtained, for retrieving existing products, approximately 68 CP seconds a day, is it worth it to make the change? The code is almost all done since I needed it for the comparison tests.

To retrieve a package located approximately in the center of the index the resources were:

	Sequential Search	Indexed Search	Difference	Difference/500
CP	29.128 SEC.	11.906 SEC.	17.000 SEC.	0.034 SEC.
MS	94.520 KON.	63.572 KUN.		
CM	4.609 KUD.	4.312 KWN.		
SR	26.284 UNS.	14.525 UNS.		
COST	2.89 $	1.59 $		

For "Not Found" products the resources were:

	Sequential Search	Indexed Search	Difference	Difference/500
CP	86.027 SEC.	48.358 SEC.	38.000 SEC.	0.076 SEC.
MS	183.127 KUN.	132.119 KUN.		
CM	4.333 KUD.	4.745 KWD.		
SR	62.003 UNS.	38.076 UNS.		
COST	6.82 $	4.18 $		

The resources utilized now by Callprg to retrieve the first and the last listed products of the index are the following:

	First Product	Last Product	Difference	Difference/500
CP	12.148 SEC.	50.170 SEC.	38.000 SEC.	0.076 SEC.
MS	62.573 KUN.	104.641 KUN.		
CM	4.311 KWD.	4.707 KWD.		
SR	14.320 UNS.	36.546 UNS.		
COST	1.50 $	4.02 $		

//////////

Default SRU Limits - An Implementation Proposal - by L. Ozga

For batch jobs at beginning of job time, every job gets an infinite SRU limit. The validated SPU limit is not checked. It is only checked when a SETASL or SETJSL card is executed. I propose adding a feature whereby the SRU limit for a batch job be set at the beginning of job to the minimum of the default and the validation limit. It should be noted that TELEX already imposes default SRU limits on its users.

This is an implementation proposal only. Since the default limit is infinite and apparently every user is currently validated for an infinite limit, this feature will not now affect any user.

However, this feature can be the mechanism for a couple of capabilities: with non-infinite limits applied, no job can get into an infinite KCL loop and a per-job "cost" limit can be applied to account numbers.

//////////

A Tale of One Machine - by J. J. Drummond

The final episode in the saga of XMIT/SEND proposals.

This proposal involves a replacement for the current SEND mechanism to enable users to send permanent files to the 6400. This proposal is a reissue of part of a previous proposal for SEND (see DSN Vol. 3, No. 12). Basically, this involves a new control statement, as follows:

SEND64(lFn=pFn,.../TY=type,UN=usernumber,PW=password)

Parameter	Explanation
type	I for indirect access, D for direct access.
usernumber	6400 user number.
password	6400 password.

This would allow a Cyber 74 or Cyber 172 user (with CXMT permission) to send permanent files to the 6400. The following restrictions also apply:

They must have a correct 6400 user number/password.
The 6400 user number must be open and have CXMT permission.
The user index of the 6400 user number must be less than AUIMAX.
The 6400 user number must be validated to create files of the specified type and size.

The file is sent to the 6400 via the TRANSIT mechanism, the validations are checked, the file saved or defined as needed, and an abbreviated DAYFILE is returned to the original sender indicating the success or failure of the transfer.

This proposal differs from the original in two respects:

1) The control statement is called SEND64 instead of SEND. This is because the original proposal allowed for permanent files to be sent to any machine and this facility is not needed on the NOS machines (with shared disk drives). Therefore, the name SEND64 more adequately reflects its function. (This will also allow SEND64 and the current SEND to coexist for a time and provide for a smoother transition.)

2) A number of parameters have been deleted. The original proposal called for additional parameters; specifically PN (packname) and MI (machine) parameters which are no longer relevant and FP(file password), M(file mode) and CT(File category) parameters which, while desirable, are not necessary in view of the small number of files being sent to the 6400 (typically, 10 to 20 files per day).

This will provide users all the capabilities of the current SEND as well as: the ability to send files to the 6400 from the Cyber 172; a DAYFILE to inform them of the results of their SEND and better enforcement of file size restrictions on the 6400.

Additionally, there will no longer be a need to send copies of the 6400 validations file to the Cyber 74 and thus a potential security problem will be eliminated.

The SEND64 control statement will create a copy of the specified file before releasing it to the queue. This means that users won't lose a file if the transfer, for some reason, fails. Also, users wishing to send a direct access file won't have to copy it to a local file first.

//////////

Multiple Copy Option - by T. J. Hoffmann

To install the multiple copy feature into COPYB, I propose to add an NC= parameter to specify the number of copies. The KRONOS N= cannot be used because CDC is already using N= for another purpose. Incidentally, NC= is what UNPAGE uses for multiple copies. Also, does this feature need to be installed into both COPY and COPYEI? COPY(X,OUTPUT,TC=I) is the same as (COPYEI(X,OUTPUT).

SYSTEM MAINTENANCE: People and Procedures

Last Week's Systems Group Meeting - by T. W. Lanzatella

1. The following proposals were discussed.

   a) Mike Frisch's proposal to straighten out the organization of files under UN=LIBRARY was approved. We agreed that because CPL is a publicly accessible file, it should be stored under the CALLPRG user number. Mike agreed that PN=UCC, UN=LIBRARY would be a suitable location for systems OPL's provided they are not public (see DSN 4, 19 p. 159).

b) Brad Blasing's proposal to add a PD option to COMPASS was approved. Brad's blanket proposal to add a PD option to any paper consuming utility was rejected.

2. Our discussion of recent security problems resulted in two resolutions.

a) Previous efforts to install password hashing should resume.

b) John Larsen agreed to begin a new charge as the UCC contact for the Malicious Users Group.

//////////

PDP 11 Communications - Part 2 - by Elie May

The second communications program is a modular communications system that may be adapted to system perturbations with minimal change. This system has been used to:

1. Send listings to 1004's.
2. Diagnose 1004's.
3. Evaluate communications performance through varying modem speeds, buffer sizes, and data encryption.
4. Communicate with the Cyber using DDCMP protocol through the RJE front end to SUPIO.

Currently this package is being used for listing and diagnosis and is being distributed to PDP 11's systems that wish to communicate with the Cyber. The executive committee has fixed a fee for the installation of this communications system. The system has been installed on PDP 11's ranging from the LSI 11 micro-computer to the PDP 11/45 mini-computer. The system takes advantage of optional hardware (KG-11 and KW11P) if in the system and the Extended Instruction Set. The program, however, can still operate if these options are not in the hardware system.

The system was designed with "tool building" (or bootstrapping), modularity, maintenance and development support concepts. It supports task activation, scheduling, task chaining, and state transition. The system has five classes of modules:

1. Executive (tasker, scheduler, console device processor).
2. Library (conversion routines, memory manager, list processing routines).
3. Operating System Interface (Current RT 2 or 3).
4. Device Support (DP-11 module and DU-11 module).
5. Information or Protocol (1004 support module and DDCMP terminal emulation module).

Modules of the same class are interchangeable. The system has a common macro library and constant definition file for ease of maintenance and adapting system to the target system's configuration.

In summary the modular communications system is providing service to Engineering, myself, and the PDP 11 user community. Its modularity, development aids, efficiency, and ability to adapt to system perturbations indicate the future potential of this system.

//////////

Reply to Steve Reisman - by C. Schofield

I don't really want to get involved in the systems/change controversy, but there are some contradictions in Steve's article of last week, and some genuine nonsense regarding MNF. Why do people get so emotional about MNF? First the contradictions:

(1) "batch of products...best I've seen" then "COBOL5...4-8 times slower than COBOL4" and "very serious...bugs". I wonder what would be the reaction to a hypothetical MNF6 which was 4 to 8 times slower than MNF5, and had new, improved, very serious bugs to boot.

(2) I don't understand how (or why) FTN and COBOL would go after (RA+65) at the same time in the same job. And we didn't know that there were "Mode 1's... you couldn't count them all" in FTN. However, it is true that there are many other things that I don't understand.

(3) Record Manager didn't save our Fortran users any time. Ask BKY, ask LOND, ask MINN, ask anybody. And, although Record Manager can apparently read any thing (my phrase), it can't even read simple little PASCAL files. And, wait for it, PASCAL runs twice as many jobs as COBOL.

Now, to quote:

> "Look at MNF. Its design goals were that it had to be the fastest, most extensive debugging compiler in the world, and it had to be compatible with FORTRAN extended's run-time package. Clearly, these are conflicting goals. Do we keep all the products stagnated at level 460 just because MNF can't adapt. Does MNF have to adapt?"

(a) The design goal of MNF was not that it be "the fastest compiler in the world". In fact, it would run quite a bit faster if certain features were removed.

(b) The design goal of MNF was to add some useful debugging - far from being "the most extensive in the world". Why not the universe?

(c) The design goal of MNF was not to interface to FTN. The older among us might indeed remember that FTN had not been invented at that time. The FTN interface was done at London, not Minnesota, and it was done about five years after MNF first appeared.

(d) Even if the alleged goals were true, I can see no reason why they are "clearly conflicting". The FTN library is great, as no doubt Steve would agree, it being part of the standard CDC system. What's "clear" about it?

(e) Who said the products should be kept at PSR460 because of MNF? I didn't say it. Jim didn't say it. Larry didn't say it. As far as I can see, the only person who said it was Steve. We have, of course, no such intention.

(f) Who said "MNF can't adapt". I didn't say it. Jim didn't say it. Larry didn't say it. As far as I can see, the only person who said it was Steve. Since MNF is currently running on everything from MACE to SCOPE 2.1.5, I would have thought it to be more adaptable than most.

(g) "Does MNF have to adapt?" No, not here it doesn't. We may have to release 472 mods for other sites. Why should we lose MSUIO? On CDC's own figures, BAMLIB is bigger and slower than MSUIO. We gain nothing. Let's have less emotion, and more facts:

(1) MNF was written as a better alternative to the RUN compiler. In this there is no doubt that it succeeded.

(2) MNF uses Record Manager with all its gee wizzery. Despite this, MNF runs jobs in 10000B words less than FTN. If you think that's peanuts, go talk to Mike Skow.

(3) The future of the operating system is not affected by MNF at all. That's why it runs on:

7600 SCOPE 2
7600 SCOPE 1
SCOPE 3.1 through 3.4.4
KRONOS 1.0 through 2.1.2
MACE, HUSTLER, IDA, UT-2D

and so on. We don't need your rotten systems.

(4) MNF runs half (HALF) the jobs at this Center. That's six (SIX) times as many as COBOL, chums. Some people might even think it's six times more important. Either way, let's have less hysteria, and leave MNF out of it.

//////////

What's All This Bruhaha? - by J. J. Drummond

Basic access method is basically a general-purpose user-oriented tool. It was written to help systems programmers and the user community to write portable software. For instance, how often has one of your working routines failed in the last year? I'm sure that basic access method could have solved the problem.

Advanced access method was designed to be a flexible facility. It was designed to enable CDC and students to write efficient programs. For instance, how many times has one of your up-and-running routines failed in the last year? It's likely that advanced access method could have significantly reduced the conversion. Advanced access method is reducing our consulting load tremendously. I'm amazed that you critics could raise your voices against advanced access method. Its time-saving documentation is great! But remember: who are the ones pushing for it? The user community.

Cyber control language has become a powerful and general package. This package was designed to help students to generate effective software. Who would ever imagine that you cretins could get so worked up about it when you don't even understand it. It's multi-purpose quality is a giant step forward out of the dark ages.

Cyber loader was designed to be a general-purpose user-oriented tool. It was created to help CDC users to write efficient programs. Cyber loader is great.

COBOL 5 was designed to be a powerful user-oriented package. It was developed and designed to help CDC users to write efficient programs. But consider this: who has the most to gain by keeping current with CDC? Our users. It is a general user aid. COBOL 5 is the best.

Common memory manager has become a general-purpose user-oriented tool. Common memory manager was developed and designed to help system analysts to write portable software. Common memory manager is, to be honest, simply amazing. It's unbelievable that you people could attack this package. Common memory manager is making the lives of many programmers far easier.

SORT/MERGE has become a powerful and general user-oriented tool. SORT/MERGE was created to enable systems programmers and system analysts to write modular programs. For example, how many times has one of your production routines failed after a minor change in the operating system? It's likely that SORT/MERGE could have solved the problem. But remember: who stands the most to gain by going with this facility? Our users.

UPDATE was designed to be a general and powerful tool. It was designed to enable system analysts and systems programmers to develop reliable software. As an example, how many times has one of your up-and-running programs ceased to function since we've converted from KRONOS to NOS? Maybe UPDATE could have avoided the problem. But condider this: who are the ones beating down our doors for this facility? Our users.

For more arguments on these (and related) topics, run the following program:

```
OLD,TEXTGEN/UN=YZE6089,PN=SPL
GET,ARTICLE/UN=YZE6089,PN=SPL
ASCII
RNH
 (*IN RESPONSE TO THE PROMPT TYPE (IN UPPER CASE):*)
ARTICLE,,XXX   (*WHERE XXX = NUMBER OF PARAGRAPHS YOU DESIRE*)

//////////
```

What's In A Name? - by J. J. Drummond

The recent articles by Andy Mickel and Larry Ozga illustrate a continuing problem in the computer industry. Namely, the day to day terminology is confusing, contradictory and often changes from year to year. Terms that were in the vogue last month are scorned this month, buzzwords are rampant and the users (as well as many professionals) are caught in the middle of this verbal vortex.

One approach is just to stick to the terminology with which most people are familiar. As the new guard joins the field they are taught the old terminology--which may no longer be completely correct but is, nonetheless, universally understood. New terms are immediately eliminated and the old dogma is modified and qualified to embrace new ideas as they emerge.

Another approach is to conduct periodic purges of the vocabulary to eliminate the words and terms that have become obsolete. Old terms (and perhaps old programmers) are then sent into the fields to be reeducated and to renew their relevance with the masses.

Both of these approaches have serious problems. With the first approach, terms are used that are not really correct for what they are describing. In the second approach, people are continually forced to learn new words and terms while existing documentation becomes obsolete.

As a third approach, I propose a solution that suffers from neither of these drawbacks. This solution can be easily developed from first principles by examining the underlying problem.

Words are an abstract representation of the thoughts or ideas they are meant to convey. If a new idea replaces an old idea then the word for the old idea can be used for the new idea if (and only if) the word is sufficiently abstract enough to represent both ideas. In other words, to prevent words from becoming obsolete, one simply has to make them abstract enough to serve more than one generation of ideas. The solution to our computer terminology problem now becomes painfully obvious. All we have to do is replace our not-nearly-so-abstract words with much-more-abstract words. As an example, I offer this extension to the terminology list printed in the last DSN.

MICKEL'S NOT SO COOL	MICKEL'S COOL	OZGA'S BEST	DRUMMOND'S ABSTRACT
Account Number	User Number	Account Number	Accounting Data
Control Card	Control Stmt.	Control Card	Control Data
Job Card	Job Statement	Job Card	Control Data
Data Cards	Data Record	Data Cards or Card Images	Data
Job Deck 7-8-9,6-7-9, 6-7-8-9 Cards	Job File EOR, EOF, EOT	Job Deck 7-8-9,6-7-9, 6-7-8-9 Cards	Data Data Partitions, End of Data
Timesharing	Interactive	Timesharing	Oneness
Telex Command	Interactive Command	Timesharing Command	Control Data
Unit Record	Line	Record	Data
Coded File	Text File	Text File or Source File	Data File
Real-Time Programming	Time Dependent Programming	Real-Time Programming	Programming
Legal	Valid, Correct	Legal	Right
Illegal	Invalid or Incorrect	Illegal	Wrong

With some reflection, it is obvious that the "abstract" terminology will stand the test of time.

//////////

Cyber Deadstart Dump Analysis from Friday, 5 October through Sunday, 23 October 1978 - by K. C. Matthews

Tuesday, 10 October

09:00 (DD-22) Cyber 74

The system hung up. There were disk error messages at control point 7. Since the crash, we have identified a problem caused by PP program 1RO (rollout job) when it encounters a disk error while writing the rollout file. Larry Ozga is investigating possible solutions. In the meantime, we eventually flawed the spot on the disk which caused this problem.

09:13 (DD-24) Cyber 74
A repeat of the previous day's problem.

We are happy to report that there were no other deadstarts during this period. There were a few other cases when the system hung up and was unavailable to users for a few minutes. For example, the DDP hung on the 172 for a while. But in these cases, the operators or systems persons were able to clear up the problem without a deadstart.

//////////

6400 Deadstart Dump Analysis (10/9 - 10/22) - by R. A. Williams

Date	Description	Tape
781011	The 7054 controller for the 844 disks on channel 6 hung with symptoms reminiscent of the hardware problem which we used to have with this equipment. To complicate things, low core was written over.	DDT-1